ALPHAI LAB 1 - DISCOVERY OF ARTIFICIAL INTELLIGENCE WITH THE ALPHAI ROBOT

This lab aims to make it clear how a machine (a robot, or a program in a more general way) is able to learn "by itself", this is called **machine learning.**

We're going to discover in particular **the reinforcement learning** and **artificial neural networks.**

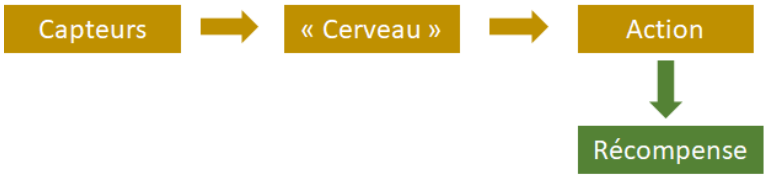
INSTALLING THE SOFTWARE

If necessary, install the software from https://learningrobots.ai/downloads/software

You will need a license activation key that will be communicated to you.

CREATE CONNECTIONS YOURSELF TO PROGRAM THE ROBOT'S BEHAVIOR.

In this first part we will see how a mini network of artificial neurons makes it possible to make action decisions based on sensor data. At the moment there is no machine learning: *you* are the one who will create the appropriate connections inside the neural network, to get the maximum rewards.



OBJECTIVE: Create the right connections to maximize the rewards and level of the robot.

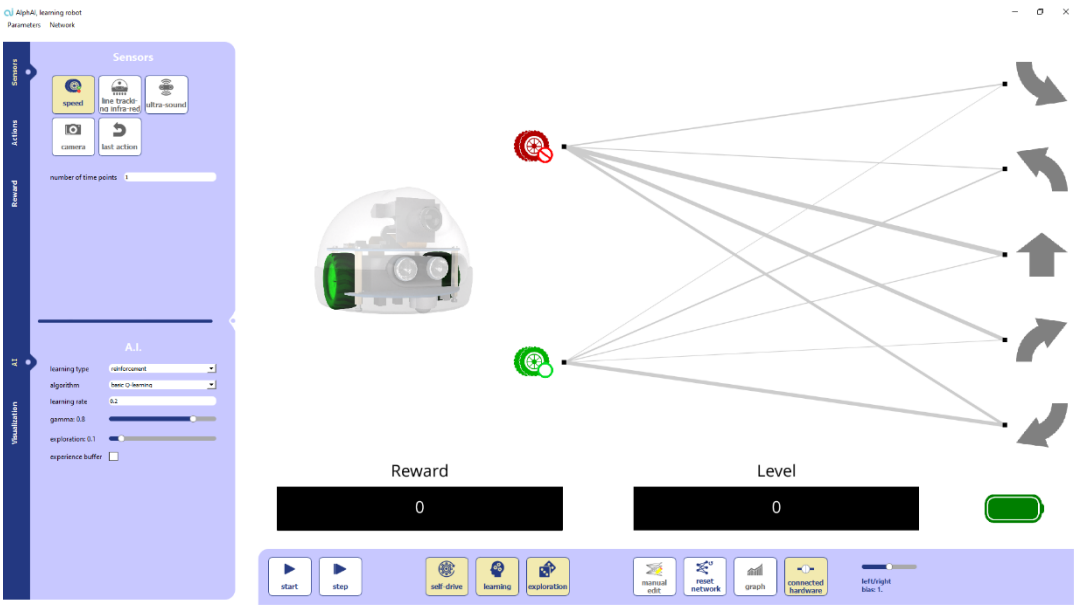
STEPS

1. Turn on a robot, write down its number below. After 30 seconds it vibrates to indicate that it is ready. It then emits a Wi-Fi network with the same number. Connect to this Wi-Fi network from the computer: the password is identical to the network name (*e.g*. ALPHAI-000123).
2. Start the program: you observe the control part on the left and bottom (in blue), the visualization part on the right (in white)
3. Press the "START" 

button: you see a small simulation robot moving lower right. To control the real robot, you have to click the “CONNECTION” button 

after stopping the simulated robot. You see its battery level appear.  You can try to roll it for a few seconds.

4. Now let's start! You're the one who's going to create the connections in his neural network:



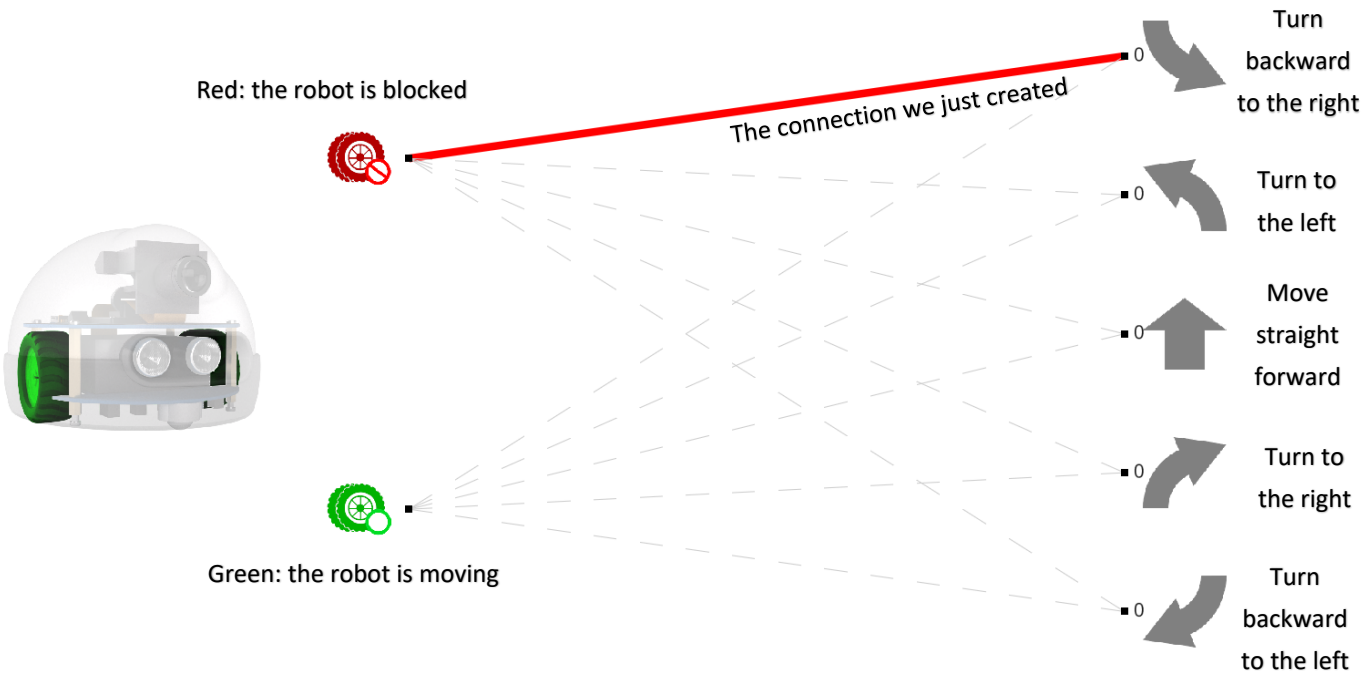
Turn off “LEARNING” and 

“EXPLORATION”, 

activate "MANUAL EDITION" and click 

“RESET A.I.” 

5. You have before you a representation of connections in the network of artificial neurons. This network has 7 neurons: 2 input neurons and 5 action neurons.



6. If you hover the dotted connections with the mouse, you can then click to **create a new connection.** These connections will transmit the activity of the neurons entering the exits; they correspond therefore to the selection of an action (on the right) when the robot is in a particular state (on the left). In the example, the connection created, will allow the robot to choose to back to the **right** when it **is blocked.**
7. Start the robot: the green neuron "I'm not stuck" lights up, and will turn on an action as soon as you have created a connection: It's up to you to play, create the right connections to allow the robot to recover the **maximum rewards.** A clue: rewards are calculated as the speed in front of the robot, but a punishment if it is blocked or goes backwards...

8. Observe the value of the level: it shows the average of the rewards received in the last 2 minutes. Note the value it achieves: we'll compare it with the level the robot can reach when it learns on its own!

OBSERVE HOW THE ROBOTTOOK IT ON ITS OWN.

Now that you've figured out how the robot has the ability to "choose" its actions (thanks to connections), let's see how the robot is able to learn what the right deeds are to choose! (always with the goal of maximizing its rewards)

OBJECTIVE: Enable the right parameters to allow the robot to learn what are the best actions.

STEPS "STEERED ROBOT":

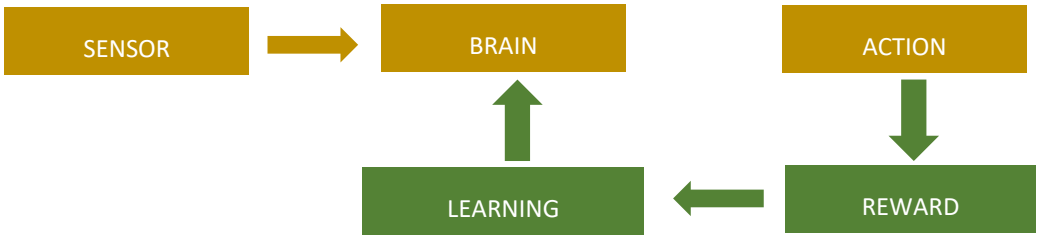
1. Turn off "MANUAL EDITION" and instead turn on "LEARNING" and then click "RESET A.I.": this puts connections at random.

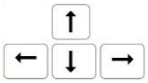
2. In the "Visualization" tab, turn on the "VALUES OF CONNECTIONS" 

display, small numbers appear above the connections. We see that the thickness of the lines symbolizes the "strength" of the connections. Can you explain why the connections displayed are not "good" and will need to be "improved" during learning?

CORRECTIONS BREAK HERE

3. For starters it is you who will pilot the robot, it is not he who will choose his actions, he will only *learn*. To do this turn off "SELF-DRIVE."

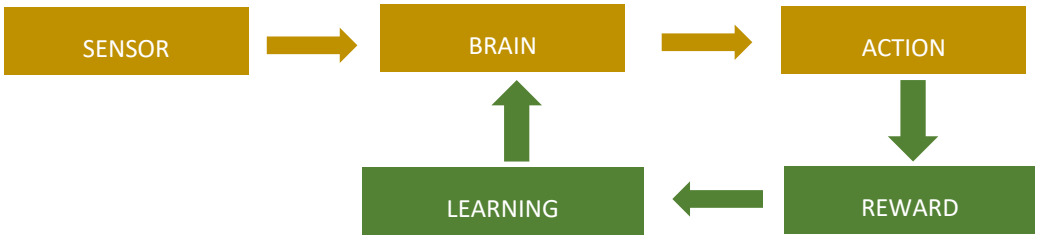


4. Click **START**. The robot is waiting for you to steer it! You can do this by clicking on the interface arrows or using the keyboard arrows. 
5. Look carefully at the thickness of the connections and the numbers that stand next to the possible actions; what do you think this number is for? Why does it change? What can be the calculation behind it?

Call us to share your thoughts! A clue: it indicates the result of **learning**...

6. When you think the learning is over, turn off "LEARNING" to stop it and reactivate "SELF-DRIVE" for the robot to behave on its own: does it behave as expected? If not, reactivate the learning to make it learn a little better...

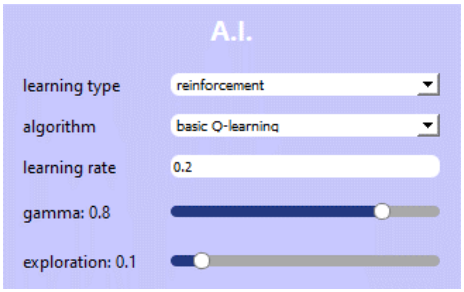
"AUTONOMOUS ROBOT" STEP



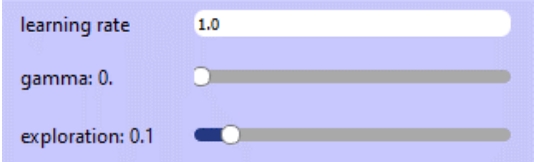
7. Start learning again this time with the robot that chooses its actions by itself (press "RESET A.I.", make sure that "SELF-DRIVE" is enabled, click "**START").** What are you observing? Is the robot learning properly? What can be missing?
8. Now also activate "**EXPLORATION**" and see how the learning goes. What does this setting do? Call us to share your thoughts.

UNDERSTAND ALL THE MATHEMATICAL DETAILS OF LEARNING:

9. You have observed that learning changes connections in the neural network. Continue the following steps to understand all the details and come up with the equation. You can also jump straight to Part III to start learning with camera and go back to that stage at the end if you have time left.
10. View the "A.I." tab: you see 3 settings, *learning* speed, *gamma* and *exploration.*



11. The *exploration* parameter represents the frequency of explorations: vary between 0 and 1 and observe the effect on the robot's behavior. Does it seem to you that the initial value of 0.1 was correct?
12. Now let's go back to something simpler: put learning speed at 1 and gamma to 0.



This comes back after one has made an action from the state to replace the connection between $a_t$and by the $s_t s_t a_t$reward received . r_{t+1}

$$Q(s_t,\ a_t) = r_{t+1}$$

Try learning with these parameter values. What's going on?

13. To avoid too much fluctuations in connections without ever converging on a stable value, reduce the speed of learning. The equation of learning then becomesα (is the parameter "learning speed").

$$Q(s_t,\ a_t) += \alpha(r_{t+1} - Q(s_t,\ a_t)\ )$$

Do we get better learning? Do you see anything that's still missing?

14. To allow the neural network to learn that when the robot is blocked the "turn around" action is better than the "straight forward" or "turn" actions, it is necessary to incorporate in the value of the actions the quality of the new state in which one arrives (here, blocked or not blocked). This is what the gamma parameter allows. The complete equation of learning becomes:

$$Q(s_t,\ a_t) += \alpha\left(\left((1-\gamma)r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1},\ a_{t+1})\right) - Q(s_t,\ a_t)\ \right)$$

Call us and ask us to help you understand it!

ADD THE CAMERA FOR U N MORE COMPLEX BEHAVIOR.

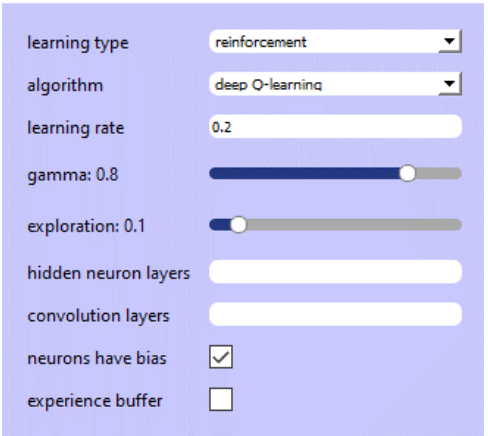
Now that you know how the robot adapts its connections to choose the most relevant actions, we're going to add a camera to the robot so it can learn to anticipate obstacles!

But be careful, before you can use the camera, you will first have to discover new parameters!

OBJECTIVE: Establish the correct settings for good learning with the camera.

"EXPERIENCE MEMORY" STEPS:

1. In the "I.A." tab, change the algorithm to "deep Q-learning," this will allow you to use multi-layered neural networks and activate new parameters.

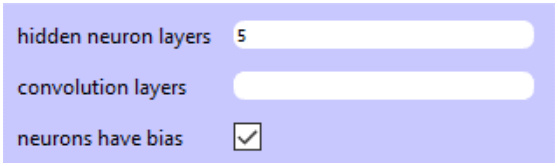


Start a new learning experience: you'll find that the connections and values of the actions can be positive or *negative* this time. Make sure a first learning is going well.

2. Turn on the "experience memory" option and restart learning. Normally it should be faster! Indeed, learning is very accelerated because the connections are changed many times per second, not according to the last action carried out only, but *to all the actions taken and rewards received* so far. So just do a straight exploration to learn that going straight is better than turning left or right.

"INTERMEDIATE NEURONS" STEPS:

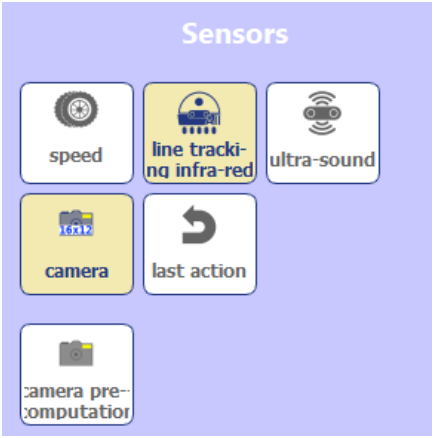
3. Add a layer of intermediate neurons and watch these new neurons and new connections appear in the interface!

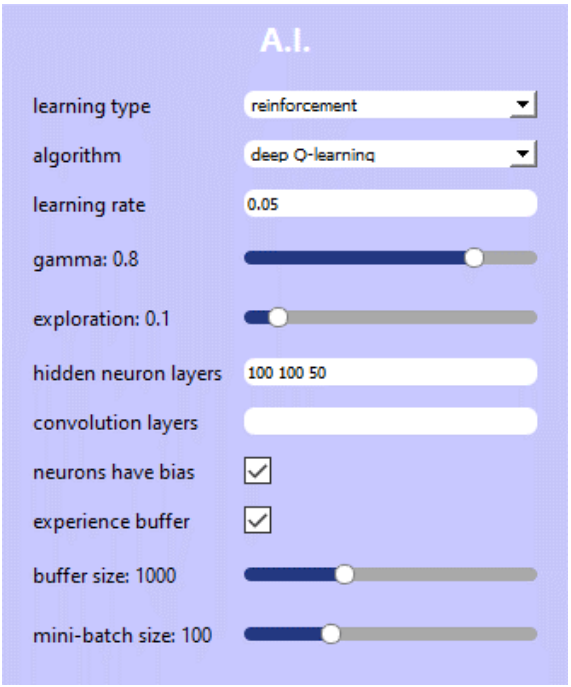


Relaunch the learning. As before, the learning algorithm learns the correct values of the actions, but the calculation of these values becomes much more complex since it depends on a large number of connections. Call us to discuss this with us.

CAMERA STEPS

4. Finally! You can turn on the camera; Choose the size 16x12 or 32x24. And here are the settings we recommend for AI: you can also try more!





5. Launch with **START** and observe the robot's learning: normally it learns successively to go straight, to turn around when it hits an obstacle, then to recognize these obstacles in advance and turn before hitting in it! This learning takes about 10 minutes. Play with visualization options to see separately the connections in the neural network, its activities, its learning...

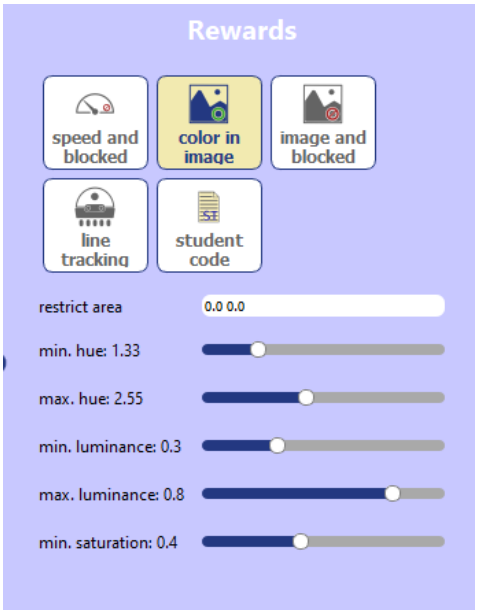
GET ALPHAI TO PLAY FOOTBALL!

You remember that the purpose of the robot is to get the maximum **rewards.** If we change how these rewards are calculated, we can make them learn something quite different!

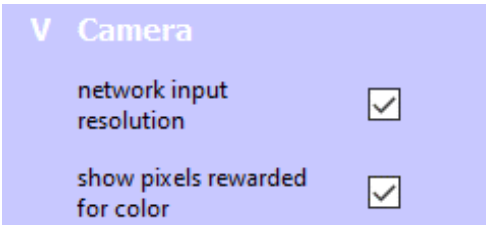
OBJECTIVE: Teach the robot to push a green balloon.

STEPS

1. Keep the settings that worked for the previous learning with the camera, and only change the method of calculating the reward!! Here are the settings to reward the robot as it advances towards the green balloon: the principle is that we detect how many pixels are green in the camera image.



2. To see the winning pixels on the screen, check "rewarded color pixels"



3. It will also be useful to decrease "loop time" to make the robot more responsive (with 0.25 it will make 4 decisions per second instead of 2)

4. You're ready! Put a green ball in the arena and start learning! This learning takes a little longer, the robot first learns to follow the ball when it is close, it takes longer to recognize it from afar.
5. The green pixels in the image may not be properly detected. In that case:
    a. Stop the robot,
    b. set **"possible actions"** in the "stop" settings.
    c. Position the robot in front of the green balloon.
    d. Start the robot! (The robot won't move though)
    e. Call us to help you adjust the settings for the color reward
6. Did you get here? You can relaunch new learnings by changing the settings you've learned !!!

CONCLUSION

Let's recap some of the concepts you've learned:

- The so-called "Artificial Intelligence" is actually poorly defined, more than a *method,* it is a *project:* that of reproducing biological intelligence !!!
- The great recent advances are in *Machine Learning:* we discovered this up close today!
- The often most effective method is the use of *artificial neural networks* that copy the activity and learning of neurons in our own brain!! Specifically, it is about increasing and decreasing the good *connections* between artificial neurons.
- As a result, some ideas in these algorithms refer to our own learning! Trials and errors; Curiosity; Learning time, Repetition and Learning during rest; Etc.
- All this is based on rigorous mathematical equations. Artificial Intelligence researchers need to be particularly qualified in *statistics ,*but also in *programming* and in intuitive *sense.*